Package 'eider'

May 13, 2024

Title Declarative Feature Extraction from Tabular Data Records

Version 1.0.0

Description Extract features from tabular data in a declarative fashion, with a focus on processing medical records. Features are specified as JSON and are independently processed before being joined. Input data can be provided as CSV files or as data frames. This setup ensures that data is transformed in a modular and reproducible manner, and allows the same pipeline to be easily applied to new data.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.1

Imports dplyr, lubridate, stringr, magrittr, jsonlite, logger, purrr, fs, tibble, rlang

Suggests knitr, rmarkdown, testthat (>= 3.0.0), tidyr

Config/testthat/edition 3

VignetteBuilder knitr

URL https://github.com/alan-turing-institute/eider

BugReports https://github.com/alan-turing-institute/eider/issues

NeedsCompilation no

Author Catalina Vallejos [ctb] (<https://orcid.org/0000-0003-3638-1960>), Louis Aslett [ctb] (<https://orcid.org/0000-0003-2211-233X>), Simon Rogers [ctb] (<https://orcid.org/0000-0003-3578-4477>), Camila Rangel Smith [cre, ctb] (<https://orcid.org/0000-0002-0227-836X>), Helen Duncan Little [aut] (<https://orcid.org/0000-0002-0897-7188>), Jonathan Yong [aut] (<https://orcid.org/0000-0002-2472-974X>), The Alan Turing Institute [cph, fnd]

Maintainer Camila Rangel Smith <crangelsmith@turing.ac.uk>

Repository CRAN

Date/Publication 2024-05-13 11:13:19 UTC

R topics documented:

`eider_example` *Obtain filepaths to example data and JSON features*

Description

Return an absolute path to the example data and JSON features provided with the package. These files are contained in the package `inst/extdata` directory.

Usage

```
eider_example(file = NULL)
```

Arguments

`file`	The filename to return the full path for. Defaults to `NULL`, in which case it will return a vector of all valid filenames.

Value

A string containing the full path to the file, or a vector of filenames

Examples

```
eider_example()
eider_example("random_ae_data.csv")
```

`run_pipeline` *Perform the entire feature transformation process*

Description

Reads in data and feature specifications and performs the requisite transformations. Please see the package vignettes for more detailed information on the JSON specification of features.

Usage

```
run_pipeline(
  data_sources,
  feature_filenames = NULL,
  response_filenames = NULL,
  all_ids = NULL
)
```

Arguments

`data_sources`	A list, whose names are the unique identifiers of the data sources, and whose values are either the data frame itself or the file path from which they should be read from. Only CSV files are supported at this point in time.
`feature_filenames`	A vector of file paths to the feature JSON specifications. Defaults to `NULL`.
`response_filenames`	A vector of file paths to the response JSON specifications. Defaults to `NULL`.
`all_ids`	A vector of all the unique numeric identifiers that should be in the final feature table. If not given, this will be determined by taking the union of all unique identifiers found in input tables used by at least one feature.

Value

A list with the following elementss:

- `features`: A data frame with all the features. The first column is the ID column, and always has the name `id`. Subsequent columns are the features, with column names as specified in the `output_feature_name` field of the JSON files.
- `responses`: A data frame with all the responses. The structure is the same as the `features` data frame.

Examples

```
run_pipeline(
  data_sources = list(ae = eider_example("random_ae_data.csv")),
  feature_filenames = eider_example("ae_total_attendances.json")
)
```

Index